GRANULATION OF SOUND IN VIDEO GAMES

LEONARD J. PAUL[1]

[1] *Lotus Audio, Vancouver, BC, Canada*
info{at}VideoGameAudio.com

Granulation of sound for games is becoming a more viable tool for sound artists as the processing power of game consoles continues to improve. Granulation is a relatively recent method of sound generation which allows for sampled sound to be modified in real-time to allow pitch to change independently of tempo among other audio effects. It has not been commonly used with the previous iterations of game platforms due to its relatively costly DSP overhead and number of voices required. With the current generation of game platforms, these limitations have been relaxed allowing for the increased possibility of real-time granulation in games. The combination of granulation and the output of a game's physics engine is a promising relationship as the physics engine can supply the large amount of real-time control data for the grain parameters that granulation requires. This paper investigates some current research in sound granulation and demonstrates some effective methods of utilizing granulation for games.

INTRODUCTION

The use of real-time granulation in games gives the video game audio artist a wide range of new techniques to fight the repetitive nature of sampled sound while maintaining the advantageous qualities of sampling's ability to capture realistic audio events and its generality of use. Examples are shown how granular synthesis could be used to augment and improve the sound for dialogue, sound effects, crowd and car engines, music and surround sound effects driven by the physics parameters of the *Unity*[1] game engine.

1 GRANULATION OF SOUND

Granular synthesis and granulation in general is a relatively recent method of real-time sound generation which allows for sampled sound to be segmented into small elements known as grains and played back in virtually limitless combinations. It has not been commonly used real-time with past iterations of game platforms due to it's costly DSP overhead and high number of voices required which often exceeded the hardware voice limit on previous consoles.

The latest iteration of game platforms (Microsoft Xbox 360, Sony Playstation 3 and Nintendo Wii) have enough audio processing power and high-resolution audio outputs to entertain the possibility of real-time granular synthesis in conjunction with gameplay. Granulation has become popular within modern audio software such as Propellerhead's *Reason*, Ableton *Live*, Native Instrument's *Reaktor, FLStudio* by Image-Line and more. There is also some recent research into the granulation related methods that are very promising methods.[2,3,4]

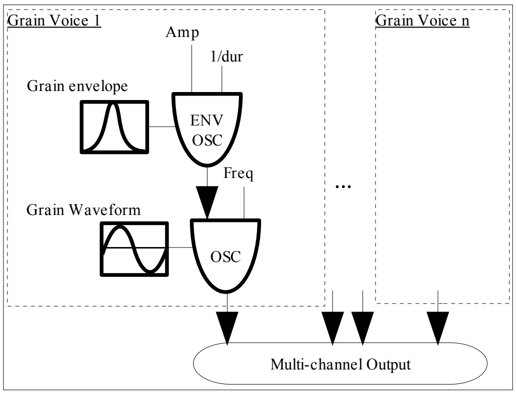

Fig. 1: *A basic granulation flowchart*

Granulation is highly capable of creating new audio textures from existing recorded sounds. If one is hoping to recreate sounds that exist in nature, then often it is best at creating textures that are similar to scraping and other forms of sounds which appear have few identifiable transient details. Granulation is a great method for creating textures which retain the spectral characteristics of the seed samples.

Some typical parameters for granulation include [5]:
1) Selection order (forwards/reverse or freeze)
2) Pitch shift (or playback rate)
3) Amplitude range
4) Spacialization/panning (static/dynamic)
5) Grain duration
6) Grain density (number of grains/second or number of grain voices)
7) Envelope (ASR shape, attack/release slope or windowing function)
8) DSP effect (reverb, filtering etc.)
9) Feedback amount (for granular delay lines)

The common difficulty when attempting to use granulation is discovering how to modulate these parameters from the physics engine to arrive at desired aesthetic result. This process requires a great degree of trial and error while listening to the results of granulation while the game is running. For granulation, the ability to rapidly prototype one's sound design is key so that the sound artist has access to a system that allows for real-time changes of the granulation parameters.

2 GRANULATION IN GAMES

Granulation can be used to dynamically change the playback speed or pitch of the speech sample independently of another in real-time. This can make the retriggering of certain speech lines (“Hey! Over here!”) sound a little less repetitive. For car engines, it has been used to add a dynamic layer over the layered dyno car engine sample loops[6]. Crowd backgrounds and chants can be made much more dynamic by variation of the voices using granulation. One the easiest uses is to augment existing non-specific backgrounds such as room tones to extend their length without noticeable loop points. Music has many possibilities as when used as part of a composition, many parameters can be changed in real-time to react to game parameters and to improve the music's responsiveness to gameplay. These cases and more will be reviewed in the real-time granular audio demonstrations that accompany this paper[7].

Even if the playback tempo (rate of moving through the source sample to select grains) and playback rate of the grains remain unchanged, the nature of random grain selection can cause a slightly different timbre due to the granulation each time the sample is granulated. This has the feeling of being slightly more organic as nature provides a certain amount of randomness, such as water flowing over rocks, to create sounds that can be listened to without the fatigue in contrast to the precise unnatural repetitiveness of sampled sound.

Sampled sound is very good at capturing a moment of time in the past but sometimes it is quite difficult to bend the sound to suit the dynamic changes of the present being simulated in real-time. One can see this as being analogous to a 2-d image that is very convincing when viewed from one angle but if the perspective changes then the illusion is lost. Similar to current 3d graphics technology, there is a combination of sampling via texturing and synthesis of using fire particles or generative fluid dynamics to simulate water. There have been investigations into using the physics of sound creation to synthesize sound directly[8] but this approach requires a separate model for each type of sound generation. It would seem natural that this approach would extend to audio as well. There has been growing interest in bringing real-time graphic generation and sound together in research which began with papers such as “Foley Automatic”[9] and continuing with research such as “Sounding Liquids.”[3] This research is perfect for games as it focuses on the real-time possibilities of audio-visual generation.

3 ADVANTAGES AND DISADVANTAGES

The difficulty of granulation in the past has been the definition of the control data for the granulation. With games, the difficulty is often more discovering intelligent ways to turn the raw data from the physics engine into numbers that control the granulation respecting which follow the aesthetic goals of the sound design.

The advantage of granulation is that it allows a high degree of control by the sound artist who produces the initial sample but wants to use granulation to reduce the static and repetitive nature of sampled sound playback. Depending on the size of the grains, one can even view delay-line granulation as a real-time digital audio effect similar to reverb or other effects. With the increase of disc space and RAM, we can make the sample resolution very high in games, but without the proper tools, these samples will become quickly repetitive and fatiguing with repeated playback. Often the most successful uses of granulation is in combination with samples as in the car example described above.

When granulation is used in games, one of the methods to reduce this possibility is to pre-segment the grains manually to taste and use a playback method to reduce stuttering such as selecting from a window to randomize from around the current playback position. Changing the grain size within an allowable range adds variation.

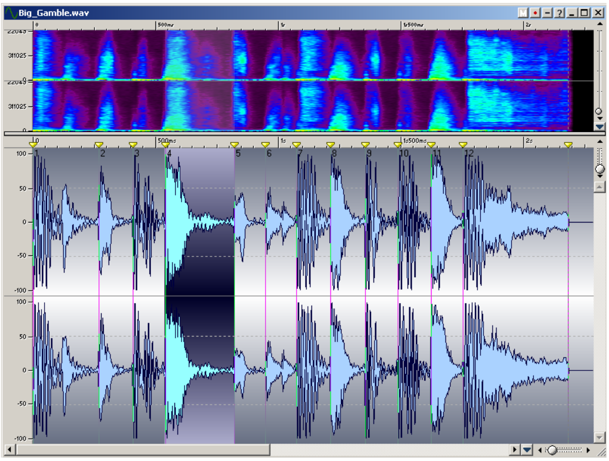

Fig. 2: *Manual grain segmentation in Wavelab*

The advantage (and disadvantage) of using granulation for time stretching is that it preserves transients (if the grains are enveloped correctly). Transients are key in our identification of a sound. Unfortunately, the granulation process often causes transients to be retriggered when stretching a sound longer, that can result in a "stuttering" sound which is easily perceived and considered unnatural. When making a sound shorter via granulation it is possible to skip important transients and produce an unconvincing result. To change the pitch of the resulting granulated signal, the playback rate of each individual grain can be modified.

When an automated method that segments grains on spectral flux boundary changes [2] to form more natural grains that already contain their own natural envelope. This method also proposed to segregate the transients and steady state portion of the grain to reduce the non-transient discontinuity between grains. When the steady state is held over adjacent grains, the transient portion is by nature discontinuous and the signal can be reconstructed with more authenticity than a more naïve method.

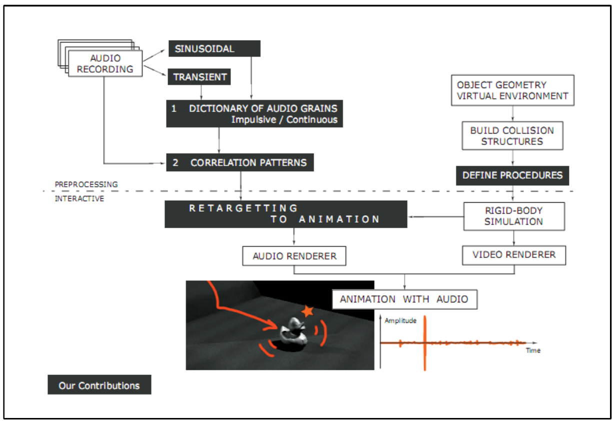

Fig. 3: *Grain analysis and decomposition into sinusoidal and transient components*[2]

4 CONCATENATIVE SYNTHESIS

Granulation can be seen as being related to concatenative synthesis [10]. Concatenative synthesis also splits an input sound into small grains, but it also categorizes them and then attempts to resynthesize an audio input stream utilizing the recorded grains and choosing the best grains to resynthesize the input. Audio manipulations such as playback rate modification can also be used to reduce the difference between the analyzed sound segment and the target grain but often a larger grain pool results is favourable as it results in less distortion of the original grain due to audio processing.

The paper "Retargetting example sounds to interactive Physics-Driven animations"[2] can be seen as an analysis of a recorded sound and then concatenative synthesis on a virtual sound for the game that is "reconstructed" by the source grains. In this method, the sound events are synthesized from the grain pool utilizing the parameters from the physics engine with some audio processing to realize the virtual sound.

5 SURROUND

Granulation is a great method for spreading a sound into a surround field. Since the sound is already being split into many small separate grains, each grain can be positioned independently to perform diffusion of the sound in surround or a multi-channel speaker array. This is particularly effective at certain ambient sounds such as water, rain and wind to give one the sense that the sound is not coming from a fixed location. In the Pure Data[11] granulation patch[12], the grains are spawned from the current position of the pointer so if the pointer is moved quickly then a strobe-like effect is produced with the oldest grains trailing the current playback location. The effect can also be used in a subtle fashion to give the effect that the sound source is from an identifiable location yet not entirely a point-source location. Stereo and multichannel sources can be used to add dynamics to the sound source and increase the sample material for the grains. In the case of a multi-channel source, the granulation can preserve the location of the grain material or mix the grain source locations together on granulation.

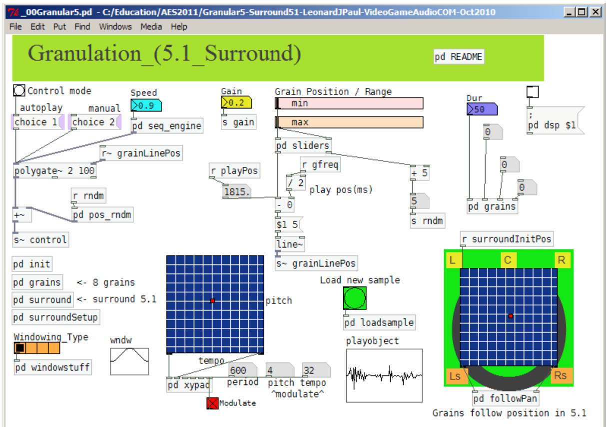

Fig. 4: *Granular 5.1 Patch in Pure Data*

6 IMPLEMENTATIONS

Fmod[13] has a granulation option in its designer tool. It requires the sound to be pre-segmented in a separate audio editor then have the pool of grains added to the sound definition. One can define if the grain selection will be entirely random with the possibility of repetition or using a shuffle function to require all grains to be played before the grain pool is refreshed. The spawn time between grain playback is also a parameter to allow grains to overlap depending on their length. It is also possible to define a scenario for concatenative synthesis by forcing playback to wait for previous grain to complete playback before starting the next grain. This granular scheme has been used in games before but it is useful to have the parameters easily controllable by the sound artist. Even in the Fmod manual, one of the difficulties of granulation implementation is the requirement that grain playback often can require an update less than 16ms which is the typical update control rate for audio for 60 frame per second games. It simply requires the update to run more frequently on a sub-frame basis. Care needs to be taken to ensure that the amount of processing per-frame total does not cause an impact to the frame rate when calculating processor intensive granulation.

7 UNITY OSC IMPLEMENTATION

The author has been utilizing the game engine *Unity* and with Open Sound Control (OSC)[14] to send the physics parameters out in real-time to code in Pure Data. In this case, Pure Data is a good prototyping environment as behavioural changes can be implemented quickly and in real-time without recompilation of the game or difficult to use in-game parameters. The ability to create behaviours that can be adjusted to suit the aesthetics of the resulting sound is also a strength as compiled languages such as C++ can not only slow the creative flow but are often not as intuitive as Pure Data for creating complex audio behaviours. It is not trivial to translate a Pure Data implementation into C++ compiled game code but gives a concise idea of the implementation for the game coder. If needed, Pure Data is open source and the core code is a BSD license which can be viewed and integrated into commercial

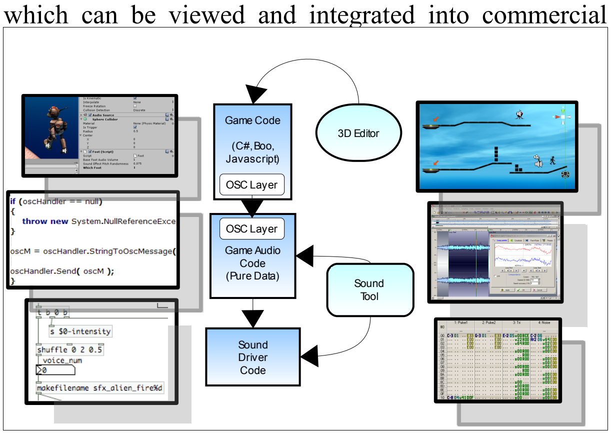

Fig. 5: *Dataflow in Unity to Pure Data patch via OSC*

Unity is a flexible game engine whose advanced physics parameters are available for free. It can be easily combined with OSC to allow it to send out real-time messages to another external program such as Pure Data. For the sake of study and research, Pure Data has been used to implement the granulation of sounds under the influence of physics parameters from *Unity* while it is running a game. In contrast to the author's previous work with OSC and Half-Life 2[18], each game must be individually integrated with the OSC code. However, the *Unity* engine is entirely free for the basic version which makes it an easier system to demonstrate to a wider audience.

The Pure Data granulation code is a basic synchronous granulation[16] of a stored sample with the capability of each of the eight grains to be placed in the surround field. Using the XY pad it is easy to dynamically change the tempo of the granulation playback position and the playback rate of the grains. Only one source sample can be used for the grains and one instance of the patch can be opened at one so advanced effects such as a granular cross-fade are not currently possible. The grains are chosen at random, so a spectral flux or other onset segmentation method could be used to require less enveloping on the grains. Currently, there are several different window types to allow for variations. Any physics parameter can be input and tied to the patch via the physics engine through OSC. This implementation allows for a rapid iteration on potential methods of interacting with the sample on a granular method with the physics parameters as input.

8 CONCLUSION

Granulation is a great additional tool for the game audio artist. It can be quite effective when used in conjunction with sample-based methods to add variation to existing playback of sample banks. For applications that have little sample memory such as the handheld game consoles, granulation can be effectively used to reduce the amount of sample data required for particular sound events. The more granulation is integrated into middleware audio solutions such as Fmod, the more rapidly the use of granulation as a generative use and as an effect will spread. The advantage of these systems is that they allow the sound artist to share their methodology used in granulation rather than be limited by their company's proprietary implementation of granulation.

The more granulation is used as a potential technique for sound design, the more its use will spread in gaming. Future research includes an implementation of the prototyping code which performs the segmentation of grains based on spectral flux boundaries and the generalization of allowing multiple instances of the patch.

REFERENCES

[1] *Unity* 3d Game Engine Framework Software. http://www.unity3d.com/.

[2] C. Picard, N. Tsingos and F. Faure, “Retargetting example sounds to interactive Physics-Driven animations,” *AES 35th International Conference, Audio in Games*, February 2009

[3] W. Moss, H. Yeh, J. Hong, M. C. Lin, and D. Manocha. 2010. “Sounding liquids: Automatic sound synthesis from fluid simulation,” *ACM Trans. Graph.* 29, 3, Article 21 (July 2010), 13 pages. http://doi.acm.org/10.1145/1805964.1805965/.

[4] C. Zheng & D. L. James, “Rigid-Body Fracture Sound with Precomputed Soundbanks,” *ACM Transaction on Graphics (SIGGRAPH 2010), 29(3), July 2009.*

[5] *Granular Synthesis for Next Generation Games.* San Jose. Game Developer's Conference. 2006. http://videogameaudio.com/GDC-Mar2006/GDC2006-GranularSynthesis-LPaul.pdf

[6] G. Durity, “Re: Vehicle Engine Sound algorithm inquiry.” http://tech.dir.groups.yahoo.com/group/gameaudiopro/message/15156/.

[7] Y. Dobashi, T. Yamamoto, and T. Nishita. “Real-time rendering of aerodynamic sound using sound textures based on computational fluid dynamics,” ACM SIGGRAPH 2003 Papers, pages 732–740, New York, NY, USA, 2003. ACM.

[8] Additional audio examples to accompany paper. http://VideoGameAudio.com/AES41

[9] K. Doel, P. G. Kry, and D. K. Pai. Foley automatic: physically-based sound effects for interactive simulation and animation. In Proc. *SIGGRAPH 2001*, pages 537–544, New York, NY, USA, 2001. ACM Press.

[10] D. Schwarz. Concatenative sound synthesis: The early years. *Journal of New Music Research*, 35(1):3–22, 2006.

[11] Pure Data Community Site and Downloads. http://www.PureData.info/.

[12] Additional Material on Granulation for Games. http://www.VideoGameAudio.com/.

[13] Fmod Audio Engine by Firelight Technologies. http://www.Fmod.org/.

[14] *Open Sound Control (OSC).* http://opensoundcontrol.org/

[15] P. R. Cook. *Real Sound Synthesis for Interactive Applications*. A. K. Peters, 2002.

[16] C. Roads (2001). *Microsound*. Cambridge: MIT Press. ISBN 0-262-18215-7.

[17] http://www.videogameaudio.com/AES-Oct2010/AES2010-Bogota-VideoGameAudio-LPaul.pdf

[18] R Adams, S. Gibson and S. M. Arisona (Eds.), *Transdisciplinary Digital Art Sound, Vision and the New Screen,* Digital Art Weeks and Interactive Futures 2006/2007 Zurich, Switzerland and Victoria, BC, Canada Selected Papers Series: Communications in Computer and Information Science, Vol. 7, 2008, IX, 501 p.